ITS/MOSY Sommersemester 2018

„Projekt: Radio-Cop“

Ein via Smartphone ferngesteuertes Fahrzeug mit Implementation eines Streams über eine bewegbare Kamera

Fakultät Design, Medien und Information
-Medientechnik-

Prof. Dr. Torsten Edeler
Prof. Dr. Andreas Plaß

Joel Ehlen	2288230
Nils Schöne	2248657
Björn Ihßen	2248303

Inhaltsverzeichnis Seite

Einleitung

Das Projekt ITS/MOSY im Sommersemester 2018 hat im Grundsatz die Entwicklung einer Virtual Reality Anwendung zum Ziel. Dabei soll ein Kamera Head auf mindestens einer Achse bewegbar sein und eine Bildübertragung per Netzwerk stattfinden. Die Kamera soll sich auf einem bewegbaren Untersatz befinden. Zur Bedienung des Fahrzeugs einschließlich Kamera dient eine Smartphone-App, die mittels Senden und Empfangen von Steuerbefehlen dem Nutzer die Kontrolle überlässt. Die Bewegungen des Fahrzeugs und der Kamera können somit über die Applikation gesteuert werden.
Das hier vorgestellte Projekt soll die oben genannten Anforderungen erfüllen. Darüber hinaus wird ein Radio implementiert. Die wichtigsten Funktionen des Fahrzeugs sind somit die Mobilität, Fernsteuerbarkeit, Bildübertragung sowie die Radio- und Musikaktivierung. In den folgenden Abschnitten wird das Projekt von der Planung über die Umsetzung bis hin zur Rezeption vorgestellt.

Planung und Konzept

Zunächst stand die Planung für das Projekt im Vordergrund. Allgemein kann die Planung (sowie die Konzeptentwicklung) dabei in drei große Bereiche eingeteilt werden. Für das Projekt sind einzelne Arbeitsschritte in den Bereichen Mechanik, Elektronik sowie Programmierung notwendig. Das Ziel der Planungsphase bestand somit darin, Einschätzungen bezüglich des Aufwands, benötigter Bauteile und der Signalverteilung in den jeweiligen Fachgebieten zu treffen. Dafür ist zu Anfang des Projekts ein Konzept entstanden, das in der sich anschließenden Arbeitsphase die Leitlinie darstellte.

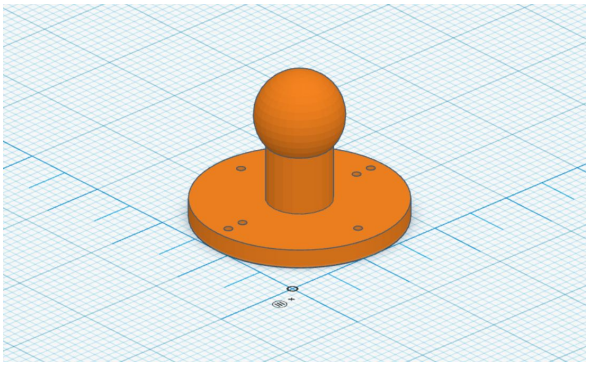
Mechanik	Elektronik	Programmierung
Karosserie des Autos	Servomotor	Schnittstelle App/Raspi
Kamera Head	Motor des Autos	Ansteuerung Motoren
Lautsprechergehäuse	FM-Empfänger	Ansteuerung Kamera
Gesamtdesign	Audioverstärker	Einbindung Live-Stream
	Tiefpassfilter	Ansteuerung Radio

In dieser Tabelle sind die einzelnen Arbeitsschritte in der Übersicht zu sehen. Im Verlauf des Projekts können diese jeweils als sogenannte Meilensteine betrachtet werden. Viele von den oben genannten Elementen sind nicht nur in einem Bereich zu verorten, sondern überschneiden sich stark. Die Tabelle zeigt jedoch die Chronologie der Planung, sowohl von links nach rechts (zunächst Fertigstellung von Mechanik und Elektronik und anschließend die Programmierung), als auch innerhalb einer Spalte (so wurden erst die Motoren des Autos elektrisch verschaltet, bevor das Radio implementiert wurde). Für weitere Details ist im Github-Repository das ursprüngliche Konzept aus dem April dieses Jahres zu finden.

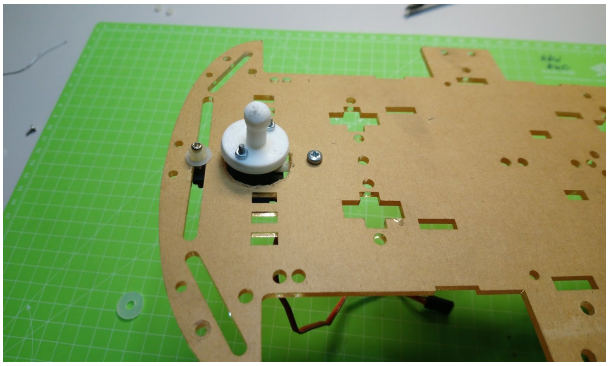
Mechanische Komponenten/ Hardware

Die Hardware des Autos ist schnell zusammengefasst. Sie besteht aus vier Rädern, die jeweils über einen DC-Motor angetrieben werden und zwei Chassis. Eines davon dient als Unterboden des Autos und das andere als das Topteil des Autos. Die beiden Chassis sind über kleine Metallstifte miteinander verbunden, sodass zwischen ihnen genug Platz für die Motoren und weitere Elektronik vorhanden ist. Neben den Motoren dient der Bauch des Autos auch als Stauraum für den Raspberry Pi sowie den 7,2V starken Akku (genauere Erläuterungen sind der Grafik im Abschnitt

Elektronik zu entnehmen). Im Bereich der Mechanik stellte sich für dieses Projekt insbesondere die Frage, wie die Bewegung der Kamera auf der Pan-Achse (links-rechts) realisiert werden sollte. Die Wahl fiel schließlich auf einen Servomotor, der die Kamera in die jeweilige Richtung bewegen soll. Um die Kamera mit dem Servomotor zu verbinden, wurde mithilfe einer 3D-Design Software (Tinkercad) ein kleines Modell entworfen und per 3D-Druck hergestellt. Das Modell besteht aus einer kleinen Plattform, die passgenau auf der Oberseite des Servomotors verschraubt wurde. Auf dieser Plattform befindet sich dann der Kamera Head, ein korkenförmiger Aufsatz, der die eigentliche Verbindung zur Webcam sicherstellt. Mit etwas Druck lässt sich die Kamera auf dem Modell befestigen. Die gesamte Konstruktion wurde abschließend im vorderen Teil des Autos platziert, indem der Servomotor von unten gegen das obere Chassis des Fahrzeugs geschraubt wurde. Für den Kamera Head ist eine entsprechende Aussparung im Chassis eingelassen.



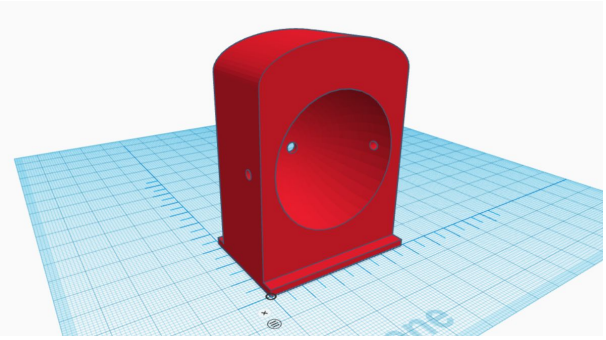
3D-Modell in Tinkercad



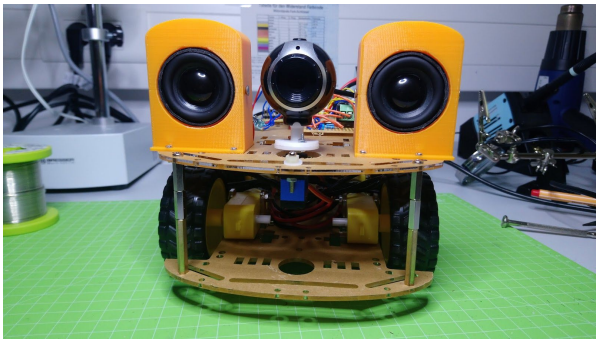
Fertige Montierung am Chassis

Ebenfalls mithilfe eines 3D-Drucks sind die Lautsprechergehäuse entstanden. Sie dienen dem festen Halt der 3 Watt starken Lautsprecher, haben allerdings auch eine kleine Resonanzwirkung durch Entstehung kleiner, sogenannter stehender Wellen im Gehäuse. Zur Befestigung am Chassis ist die Grundfläche der Gehäuse nach außen etwas vergrößert worden, Schrauben mit zwei Millimeter Durchmesser stellen die Verbindung zum Chassis her. Im Gehäuse selbst sind die Lautsprecher mit zwei seitlichen Schrauben fixiert. Die obere abgerundete Form dient

vorwiegend dem Design. Insgesamt sind die Lautsprechergehäuse leicht hinter dem Kamera Head auf dem Chassis des Autos positioniert, um das Sichtfeld der Kamera bei deren Bewegung nicht einzuschränken.



3D-Modell des Gehäuses

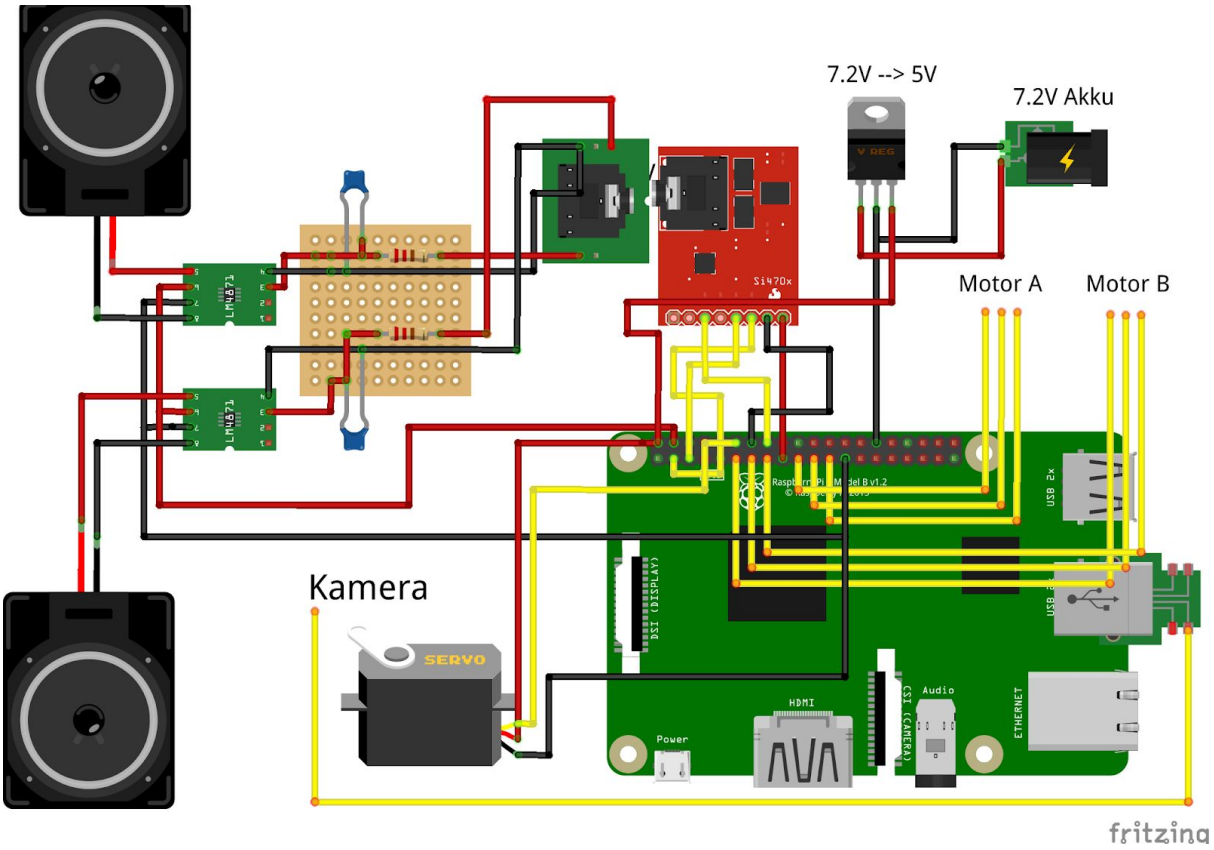


Finale Position auf dem Auto

Im hinteren Bereich des Autos befindet sich die zentrale Platine, an die auch das Flachbandkabel des Raspberry Pi angeschlossen ist. Um dieses Kabel in den Bauch des Autos zu führen wurde mithilfe eines Dremels eine entsprechende Aussparung geschaffen. Das Kabel reicht dann bis zum vorderen Teil des Autos, wo der Raspberry seinen Platz hat.

Elektronik

Der Radio-Cop verfügt über viele Funktionen, bei denen sowohl die Stromversorgung als auch der Signalfluss gewährleistet sein muss. Die zentrale Steuereinheit stellt dabei der Raspberry Pi dar, dessen Signale über das oben bereits erwähnte Flachbandkabel zum hinteren Teil des Autos zu einer Platine geleitet werden. Von dort verteilen sich die Signale sternförmig zu den jeweiligen Komponenten. Die folgende in Fritzing erstellte Grafik zeigt schematisch die elektrische Verschaltung des Radio-Cops (Hinweis: Die Verschaltung der DC-Motoren über eine sogenannte H-Brücke wurde von einem Projekt aus dem letzten Semester übernommen und wird in dieser Grafik daher lediglich angedeutet).



Elektrische Verschaltung des Radio-Cops. Die Signale für die DC-Motoren sind hier nur angedeutet und laufen anschließend über eine H-Brücke. Rote Leitungen implizieren Spannung, schwarze Leitungen Masse und gelbe Leitungen weisen auf Signalübertragungen hin.

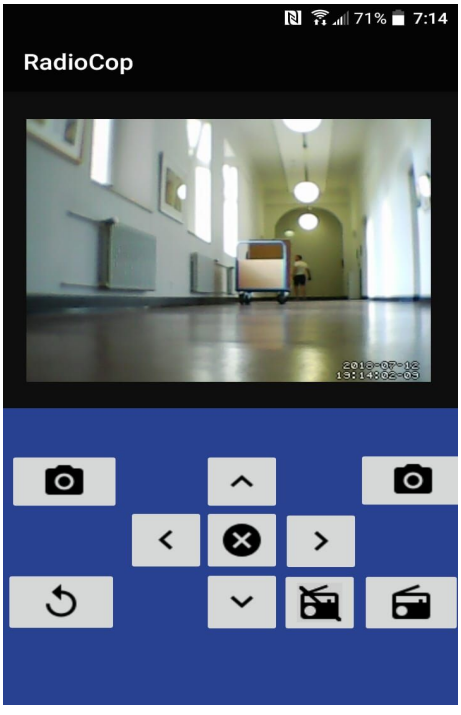
Der Radio-Cop wird über einen 7,2V starken Akku gespeist. Um die Spannung auf die maximalen 5V des Raspberry Pi abzusenken, ist der Akku an einen Spannungswandler angeschlossen. Die Kamera wird über USB mit dem Raspberry verbunden.
Der Servomotor ist an GPIO-Pin 12 angeschlossen und wird mit 5 Volt vom Raspberry Pi versorgt. Die Bewegungen des Servos werden über eine Pulsweitenmodulation (PWM) realisiert (vgl. Code). Um das Radio zu implementieren, ist ein sogenanntes FM-Empfängermodul (Si470x) an den Pi angeschlossen. Es benötigt 3.3 Volt Spannung. Für die Signalübertragung werden der Serial Data Pin (SDA), der Serial Clock Pin (SCL) zur Taktübertragung sowie ein GPIO-PIN (in diesem Fall GPIO23) für die Reset-Funktion des Radios genutzt. Das Modul wird über den Raspberry Pi

über die sogenannte I2C-Schnittstelle angesprochen. Die eigentliche Signalübertragung zum Modul erfolgt dabei synchron über die SDA- und SCL-Leitungen.
Am Ausgang des Si470x liegt ein Audiosignal in Form einer Klinkenbuchse (3.5 Millimeter) an. Mithilfe eines Split-Steckers können die drei Adern des Klinke-Signals direkt abgegriffen werden. Um das Signal auf die Lautsprecher zu übertragen wird ein zusätzlicher Audioverstärker benötigt, da es sich um ein passives System handelt. Weil das Audiosignal in Stereo vorliegt, ist auch der Audioverstärker ein 2x3 Watt starker Stereo Verstärker (der Stereo Verstärker des Projekts ist in der Bauteilbibliothek von Fritzing nicht enthalten, sodass in der obigen Grafik zwei Mono Verstärker abgebildet sind). Der Amp selbst wird mit 5 Volt Spannung versorgt und gibt die verstärkten Signale dann an die Lautsprecher weiter.

Im Verlauf des Projekts hat sich gezeigt, dass der Audioverstärker anfällig für die hohen Frequenzen des FM-Moduls ist. Aus diesem Grund ist dem Audioverstärker ein selbst entworfener und gebauter Tiefpassfilter vorweg geschaltet, der auf eine Grenzfrequenz von etwa 16 kHz dimensioniert ist. Es handelt sich für beide Kanäle (links und rechts) jeweils um einen Tiefpass 1. Ordnung, realisiert durch ein einfaches RC-Glied. Die Kondensatoren besitzen dabei etwa 100 pF und die Widerstände je 100 Ohm. Dadurch konnten die hochfrequenten Störgeräusche deutlich abgesenkt und die Audioqualität erhöht werden.
Um die FM-Signale zu empfangen, dienen die Lautsprecher direkt als Antenne. Die jeweilige Trägerfrequenz wird softwareseitig festgelegt. Aufgrund der relativ kurzen Wellenlängen (rund drei Meter) im Verhältnis zu den Sendestrecken, ist die Position des Radio-Cops sowie die jeweilige Sendeleistung des Radiosenders entscheidend für die Empfangsqualität.

Programmierung

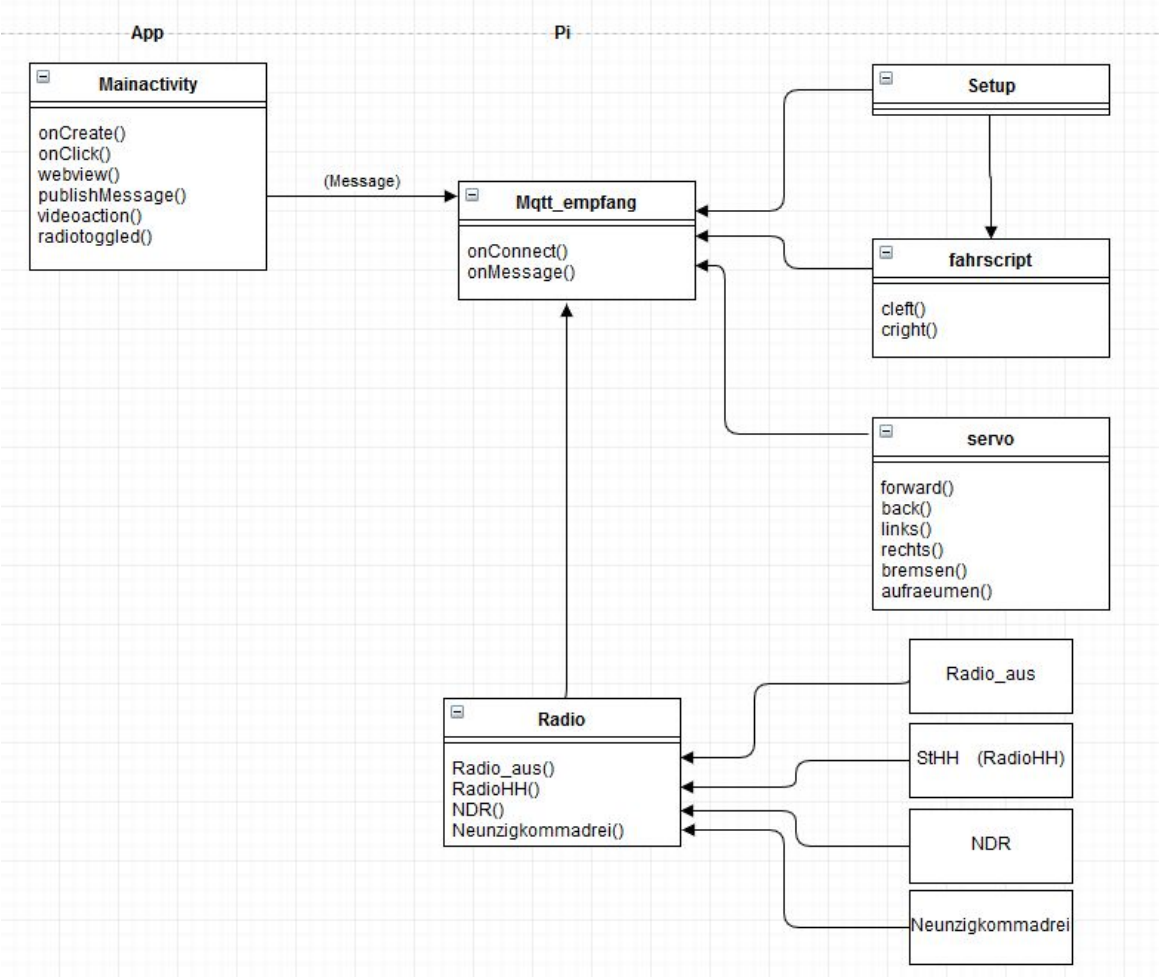
Die Programmierung des Radio-Cops besteht aus zwei Teilen: Einer App und Skripts im Pi. Beide Teile sind dabei durch ein ein Mqtt-Protokoll über die Seite von Hivemq verbunden.
Die Radio-Cop App besteht funktional hauptsächlich aus Anbindungen der Buttons an die publishMessage()-Funktion, welche eine voreingestellte Nachricht an den Mqtt-Broker. Zuvor werden sämtliche Buttons über einen switch in der onClick()-Funktion differenziert. Die einzigen Besonderheiten sind der Reload- und der Radiobutton. Der Radiobutton wird in einer zusätzlichen Funktion in einer If-Schleife aufgeteilt. Jeder Schleifenteil verändert eine Variable, welche als Counter benutzt wird und als Bedingung für die If-Schleife steht. Durch die If-Schleife wird bestimmt, welcher Radiosender schlussendlich am Pi abgespielt wird.
Zusätzlich ist zu erwähnen, dass der Button für die Ausschaltung des Radios, wenn man ihn bei Aktivität des Radios drück auch den Radiosender zurücksetzt, damit der User nicht wieder umschalten muss um seinen Sender später weiterzuhören. Der Reloadbutton lädt simpel den Link der Webview erneut um möglichen Anzeigefehlern entgegen zu wirken.
Designmäßig sieht die App wie folgt aus.

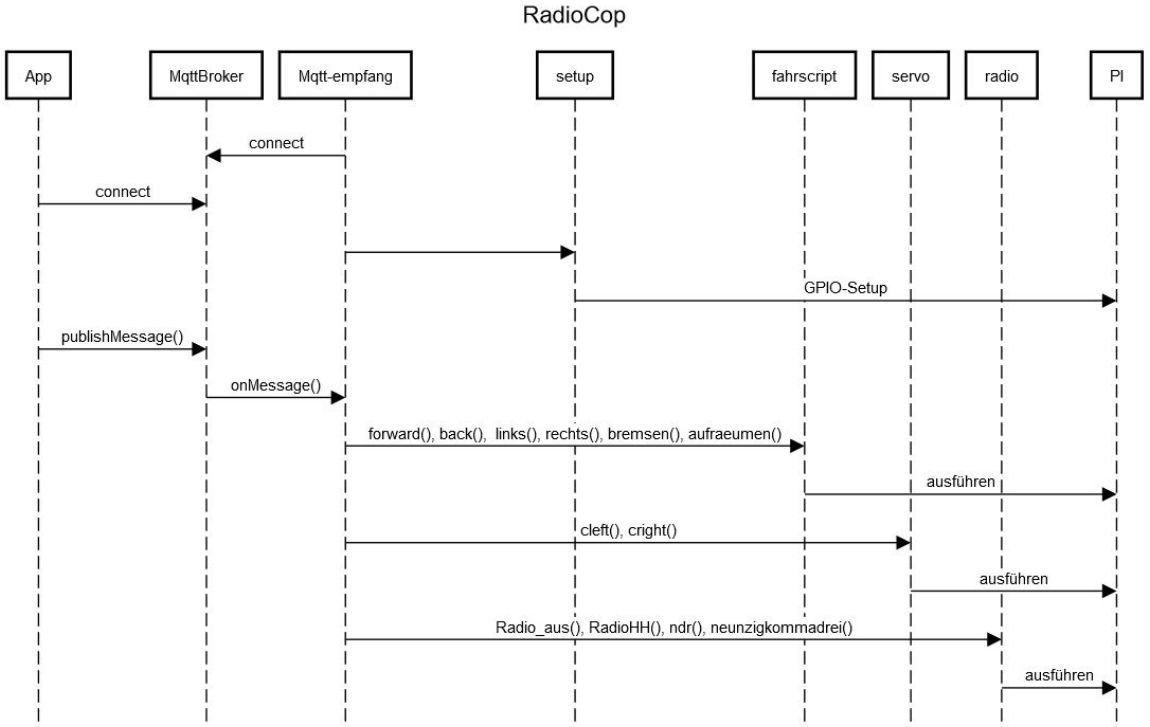


Die Webview nimmt die obere Hälfte der App ein und die Buttons die untere Hälfte. Die Buttons sind an Guidelines orientiert und lassen durch die Anordnung und ihrem Bild erschließen, welche Funktion, welcher Button besitzt. Die App wurde mit Androidstudio erstellt und ist in Java programmiert.

Nachdem die App eine Nachricht an den Mqtt-Broker zu einem festgelegten Topic geschickt hat, wird diese vom Pi, welcher das Topic subscribed hat, wahrgenommen. Das Skript “Mqtt_Empfang” bemerkt die Nachricht und leitet sie je nach Inhalt weiter und führt die zugehörige Funktion aus. Die Funktionen sind in drei Skripten aufgeteilt. Das Skript ”Radio” beinhaltet sämtliche Radioausführungen, “fahrscript” alle Funktionen zur Fahrzeugbewegung und “Servo” alle Funktionen zur Bewegung des Servomotors, an welchem die Webcam montiert ist. Das“Radio.py” Skript führt im Gegensatz zu den Anderen ein C++ Skript in einem anderen Ordner auf zur Senderwahl.
Dies liegt daran, dass unsere Radiofunktionen durch das Repository “SI4703_DRIVER” von “Tomtombusiness” inspiriert sind. Die zwei SI4703_Breakout Skripte haben wir dabei übernommen. Die Skripte, welche den Radiosender bestimmen, haben wir verändert, damit der Pi keine Verzögerung bekommt (ursprünglich der Fall) und um spezifische Radiosender zu empfangen. Da die übernommenen Radioskripte mit C++ geschrieben sind, sind auch unserer Senderskripte in dieser Sprache geschrieben .
Das “Servo.py”-Skript steuert den Servo und dadurch die daran montierte Kamera. Je nach Richtungstaste der Kamera bewegt sich der Servo. Er bewegt sich dabei zwischen neun festgelegten Positionen hin und her. Durch die Variable c erkennt das Programm, an welcher Position die Kamera sich nun befindet und auf welche zwei möglichen Positionen er sich bewegen kann. Zum Bewegen des Servos wurde die bereits vorhandene PWM() Funktion benutzt.
Das “fahrscript.py”-Skript schaltet die Räderdrehrichtungen und die Geschwindigkeit um das Auto in eine bestimmte Richtung fahren zu lassen. Dabei sei noch besagt, dass das “Setup.py”-Script zuvor die GPIOs der Motoren als Ausgang gesetzt hat und aktiviert hat. Die Skripte im Raspberry Pi sind in Python geschrieben. Die Radio Anbindung ist in C++ geschrieben.

Diagramme


App
Pi
Mainactivity
onCreate()
onClick()
webview()
publishMessage()
videoaction()
radiotoggled()
(Message)
Mqtt_empfang
onConnect()
onMessage()
Setup
fahrscript
cleft()
cright()
servo
forward()
back()
links()
rechts()
bremsen()
aufraeumen()
Radio
Radio_aus()
RadioHH()
NDR()
Neunzigkommadrei()
Radio_aus
StHH (RadioHH)
NDR
Neunzigkommadrei


RadioCop
App
MqttBroker
Mqtt-empfang
setup
fahrscript
servo
radio
PI
connect
connect
GPIO-Setup
publishMessage()
onMessage()
forward(), back(), links(), rechts(), bremsen(), aufraeumen()
ausführen
cleft(), cright()
ausführen
Radio_aus(), RadioHH(), ndr(), neunzigkommadrei()
ausführen

Fazit

Das Projekt ist insgesamt gelungen. Der Radio-Cop erfüllt die zu Beginn geforderten Funktionen und bietet darüber hinaus noch die Möglichkeit ein Radio zu aktivieren. Das Fahrzeug reagiert mit sehr geringen Latenzen auf die jeweiligen Eingaben des Nutzers (bei entsprechend ausreichender Datengeschwindigkeit des Netzwerks) und lässt sich somit sehr komfortabel über das Smartphone steuern.
Im Test mit vielen unterschiedlichen Nutzern schien die Bedienung des Kontrollpanels anfangs etwas schwierig, doch nach einer kurzen Eingewöhnungsphase war die Mehrzahl mit dem Handling vertraut. Die Aktivierung des Radios und die Möglichkeit verschiedene Radiosender auszuwählen verursachte im Praxistest einen besonderen Spaßfaktor.

Unabhängig davon lässt sich der Radio-Cop natürlich auch noch erweitern, indem beispielsweise der Umfang der Funktionen erhöht wird. So ist die Kontrolle über die Lautstärke des Radios eine sinnvolle Ergänzung. Auch die Anpassung der Geschwindigkeit des Radio-Cops stellt eine mögliche Erweiterung dar. In Bezug auf die Grundfunktionen könnte vor allem die Bewegung der Kamera optimiert werden. Wünschenswert ist eine flüssigere Bewegung, sodass bei der bei Betätigung der entsprechenden Buttons auf dem Smartphone kontinuierliche Werte an den Servomotor übergeben werden.